Cork. For example the label fields may then contain : first letters of firstname, surname, address1 and address 2.

The grouping criteria may then be set to: X(2 to 4) number of common labels. Matching is only carried out on records whose label fields contain two or more of the same letters. The keyletter may also be derived from the soundex fields.

In many cases keyletter may not be the appropriate labelling routine. The grouping module must have the flexibility to allow the user to define a number of bespoke labelling routines appropriate to the dataset (for example – if a particular data element within a dataset has a particularly high confidence level, grouping may be focused largely on this). He may do this by:

a. selecting a default grouping configuration predefined for this type of dataset,

b. firstly selecting the most appropriate fields, secondly selecting the appropriate labelling routines from a menu, thirdly defining the grouping criteria for the labels, or

c. as above but inputting customised labelling routines.

Example

Input Record:

Firstname	**Surname**	**Address1**	**Address2**	**Address3**	**DOB**	**Telephone**
John	O'Brien	3 Oak Rd.	Douglas	Co. Cork	20/4/66	021-234678

Output Record

FN_stan	**FN_Soundex**	**FN_Root**	**SN_stan**	**SN_Soundex**	**SN_root**	**A1_Num**
John	Jon	Jonathon	OBrien	O-165	Brien	3
A1_text	**A1_text_soundex**	**A1_st**	**A2_text**	**A2_text_nysiis**	**A3_st**	**A3_text**

Oak	O-200	Road	Douglas	DAGL	County	Cork
DOB_Eur	**DOB_US**	**Telephone**	**Tel_local**			
20041966	04201966	35321234678	234678			

Output Record Grouping Labels

FN_keyletter	SN_keyletter	A1_keyletter	A2_keyletter	A3_keyletter
J	B	O	D	C

Similarity Vector Extraction Module 12

Each data field within a record is compared with one or more fields from the other record of a pair being compared. All records in each group are compared with all of the other records in the same group. The objective here is to ensure that equivalent data elements are matched using an appropriate matching routine even if the elements are not stored in equivalent fields.

Each pair of records is read into the vector extraction module from the preprocessed datafile. This module firstly marks the data fields from each record which should be compared to each other. It then carries out the comparison using one of a range of different string matching routines. The string matching routines are configured to accurately estimate the "similarity" of two data elements. Depending on the type/format of the data elements being compared, different matching routines are required. For example, for a normal word an "edit distance" routine measures how many edits are required to change one element to the other is a suitable comparison routine. However for an integer it is more appropriate to use a routine which takes into account the difference between each individual digit and the different importance level of the various digits (i.e. in number 684 the 6 is more important than the 8 which is more important than the 4). Examples of matching routines are edit distance, hamming distance, dyce, and least common substring routines.

The output of the matching routine is a score between 0 and 1 where 1 indicates an identical match and 0 indicates a definite nonmatch. The output of a data field scoring routine is a set of *similarity scores* one for each of the datafield pairs compared. This set of scores is called a *similarity vector*.

The module 12 allows the user to select the data fields within the dataset/(s) to be used in the matching process, to select which fields are to be matched with which, and to define the matching routine used for each comparison. The user configures the process by:

- selecting from a menu of default configurations suitable for the dataset(s),
- manually selecting the data fields to be compared and selecting the appropriate matching routine from a menu of predefined routines, and
- manually creating customised matching routines to suit particular data field types.

Example

Input Record 1

FN_stan	**FN_Soundex**	**FN_Root**	**SN_stan**	**SN_Soundex**	**SN_root**	**A1_Num**
John	J-500	Jonathon	OBrien	O-165	Brien	3
A1_text	**A1_text_soundex**	**A1_st**	**A2_text**	**A2_str_soundex**	**A3_st**	**A3_text**
Oak	O-200	Road	Douglas	D242	County	Cork
DOB_Eur	**DOB_US**	**Telephone**	**Tel_local**			
20041966	04201966	35321234678	234678			

Input Record 2

FN_stan	FN_Soundex	FN_Root	SN_stan	SN_Soundex	SN_root	A1_Num
Jon	J-500	Jonathon	Bryan	B-650	Brien	-
A1_text	**A1_text_soundex**	**A1_st**	**A2_text**	**A2_text_sdx**	**A2_st**	**A3_text**
Oakdale	O-234	Close	Oake	0-230	Road	Duglass
A4_st	**A4_text**	**A4_text_sdx**	**DOB_Eur**	**DOB_US**	**Telephone**	**Tel_local**
County	Cork	C-620	02041968	04021968		

Output Similarity Vector

FN_stan	FN_Root	SN_stan	SN_root	A1_Num	A1_text	A1_st
.7	1	.5	1	.5	.5	0
A2_text	**A2_st**	**A3_text**	**A4_st**	**A4_text**	**A1A2_text**	**A2A1_text**
0	0	0	0	0	.8	0
A2A3_text	**A3A2_text**	**A3A4_txt**	**DOB_Eur**	**DOB_US**	**Telephone**	**Tel_local**
.8	0	1	.8	.8	-	-

The output of the data field matching process is a vector of similarity scores indicating the similarity level of the data fields within the two records. The data field matching module is capable of doing a user-defined number and type of comparisons between two data records and generating a score for each – i.e. the user will define which fields / elements of one record will be compared to which elements in the other record. The user will also define which matching algorithm is used for each comparison. In defining these parameters the user can:

- Select a default matching configuration stored in the system 1 for a specified field type.

- Select the required matching routine for a particular data field type from a menu of predefined routines.
- Input a customised matching routine

Data Record Scoring Module 13

The aim of the data record scoring is to generate a single similarity score for a record pair which accurately reflects the true similarity of the record pair relative to other record pairs in the dataset. This is done by using a variety of routines to compute a similarity score from the similarity vector generated by the module 12.

There are two different types of routine used by the module 13 to generate a score.

- Rule-based routines – these routines use a set of rules and weights to compute an overall score from the vector. The weights are used to take into account that some fields are more indicative of overall record similarity than others. The rules are used to take into account that the relationship between individual field scores and overall score may not be linear. The following is an example of a rule based computation.

  FN = Largest of (FN_stan ,FN_Root)
  SN = Largest of (SN_stan, FN_Root)
  A1_text = Largest of (A1_text, A1A2_text)
  A2_text = Largest of (A2_text, A2A1_text, A2A3_text)
  A3_text = Largest of (A3_text, A3A2_text)
  DOB = Largest of (DOB_Eur, DOB_US)
  Score = FN + SN +A1_text+A2_text+A3_text+A4_text+ (A1st+A2st+A3st+A4st)/4

- AI based routines – these routines automatically derive an optimum match score computation algorithm based on examples of correct and incorrect matches identified by the user. Depending on the situation – the type of AI technology used may be based on either neural networks or case based reasoning.

The optimum routine required to derive the most accurate similarity scores for all record pairs are highly specific to the types and quality of data within a particular dataset. For this reason default routines generally do not give the best match accuracy. In order to achieve top levels of accuracy, a trial and error process is implemented by the tuning manager 4 to "tune" the scoring routine. This requires the user to:

- run the whole matching process a number of times for a portion of the dataset.
- inspect the results after each run to check the proportion of correct and incorrect matches.
- manually adjust the parameters of the score computation routine.

This process is difficult to do with a rule based routine as there are a large number of variables to tweak. However the AI based system is ideal for this process. It removes the need to tweak different variables as the AI technology derives a new score computation routine automatically based on the learning from the manual inspection of the match results. Since the AI process requires training data, the system 1 uses a rule based routine on the first training run and uses an AI routine thereafter.

The record scoring module 13 is configured to allow user selection or setup of both the rules based and AI-based routines. The user configures the rule based routine by:

- Selecting from a menu of rule-based routine configurations predefined for common dataset types.
- Selecting a predefined configuration but adjusting individual parameters (e.g. weighting of a certain field type).
- Defining a customised routine.

The user will setup the AI based routine by:

- Selecting a recommended AI-based routine for the particular matching conditions (one-off batch matching, ongoing periodic matches etc.)

- Selecting from a menu of configurations of that AI-based routine predefined for common dataset types.
- Selecting a predefined configuration but adjusting individual parameters.

It will be appreciated that the system achieves fast and easy set up and configuration of new matching processes involving new datasets or match criteria, and easy set up of adhoc matching analyses. The system also achieves scheduling of ongoing periodic matching processes using predefined configurations. The system is callable from third party applications or middleware, and it has the capability to read data from a range of input data formats, and to provide a range of output data functions.

Important advantages of the system are:

1. Accuracy. It is capable of delivering highly accurate automated matching through the use of complex layers of processing and matching routines to compensate for the full range of data matching problems. It minimises the number of true matches not identified and non-matches labelled as matches.
2. Configurability. It enables easy setup of customised routines often required due to the highly specific nature of individual datasets. It allows the user to select parameters based on knowledge of which fields are likely to be most indicative of a match, likely quality of individual fields, and likely problems with fields/elements. The system 1 uses "wizard" type process to help the user to configure bespoke routines to remove problem characters within fields, and transform elements into standardised formats.
3. Ease of set up. There is built-n intelligence to facilitate high accuracy set up and tuning by a non-expert user. Setup is based on users knowledge of the data, and it guides user on development of processing routines. Articial intelligence is used to automatically tune the matching process based on examples of good and bad matches as verified by user.

4. Speed: It uses intelligent processing to quickly reduce a dataset to a subset of "all possible matches". The high-speed pipeline 6 maximises processing speed.
5. Open Architecture. The architecture uses component – based design to facilitate easy integration with other systems or embedding of core engine within other technologies.

The system of the invention is therefore of major benefit to businesses by, for example:

improving the value of data so that it is business-ready;

reducing project risks and time overruns in data migration projects; and

reducing manual verification costs.

The system is also very versatile as it may interface on the input side with any of a wide range of legacy systems and output cleaned data to a variety of systems such as CRM, data-mining, data warehouse, and ERP systems. Furthermore, the structure of the system allows different modes of operation including interactive data cleaning for data projects, batch mode for embedded processes, and real time mode for end-user applications.

The invention is not limited to the embodiments described but may be varied in construction and detail.